

RJMCMC based Text Placement to Optimize Label Placement and Quantity

Guillaume Touya, Thibaud Chassin

Univ. Paris-Est, LASTIG COGIT, IGN, ENSG, F-94160 Saint-Mande, France; firstname.lastname@ign.fr

Abstract: Label placement is a tedious task in map design, and its automation has long been a goal for researchers in cartography, but also in computational geometry. Methods that search for an optimal or nearly optimal solution that satisfies a set of constraints, such as label overlapping, have been proposed in the literature. Most of these methods mainly focus on finding the optimal position for a given set of labels, but rarely allow the removal of labels as part of the optimization. This paper proposes to apply an optimization technique called Reversible-Jump Markov Chain Monte Carlo that enables to easily model the removal or addition during the optimization iterations. The method, quite preliminary for now, is tested on a real dataset, and the first results are encouraging.

Keywords: label placement, optimization, RJMCMC, cartography

1. Introduction

Text placement is one of the longest tasks in map design, and many proposals have been made to automate the process with optimization techniques (see an exhaustive review in Rylov & Reimer 2014). These methods are now quite effective, and they turn common quality criteria such as preferences on text position around the symbol, text overlaps with other map elements, or ambiguity between two labels, into numerical equations that are inserted into classical meta-heuristics such as simulated annealing to find the optimal placement. But most of existing techniques have one common drawback: the amount of labels to display is fixed at the beginning of the optimization. The optimized decision is the best position for the label, the label is never removed, even if there is no good location in the map. This paper proposes to use an optimization technique called RJMCMC for “Reversible-Jump Markov Chain Monte Carlo” (Green 1995), able to include text placement and presence into the optimization. The second part of the paper briefly describes how we modelled label placement as a RJMCMC optimization, and the third part presents some preliminary results.

2. Description of the Optimization Method

The family of Reversible-Jump Markov Chain Monte Carlo optimization methods is composed of stochastic methods with a varying dimension solution space. Usually, stochastic optimization methods such as simulated annealing that was widely used in label placement (Barrault 2001), pick a label to displace (with varying randomness) and evaluate the global satisfaction of the system with this new location for the label, and the systems stops when an optimal solution is achieved. The evaluation is made with a cost or energy function that summarizes label position and the conflicts with other labels. The solution space is very large, and if we use the standard 8-position model for one label (see Fig. 1), it counts 8^n configurations where n is the number of labels. With RJMCMC, we add, at each iteration, the possibility of “birth” or “death” for any label, reducing or increasing the dimension of the solution

space. The global quality of the output is computed after each iteration, based on an energy function to minimize (Eq. 1).

$$(Eq. 1) \quad e_{total} = e_{unary} + e_{binary} + e_{collection}$$

The unary energy sums for each label the satisfaction energy related to constraints that only affect one label, e.g. the position of the label around the point feature it labels. The binary energy sums for each label the satisfaction energy related to constraints of one label with its neighbors. For instance, the possible overlaps of two labels are evaluated by the binary energy. Finally, the collection energy is related to constraints on groups of labels, which allows, for instance, the preservation of density differences in the map. The use of three types of energies makes the process faster as when one label is displaced only the unary of this label, and the few binary and collection energies that involve this label are recomputed.

We used the different criteria listed by Rylov & Reimer (2014) that summarize the existing literature: position around the point (adapted here with continuous positions, Fig. 1 and Eq. 2) as the unary energy, overlaps with map symbols, ambiguity with nearby labels as binary energies, and label density as a collection energy. In order to make the system remove some labels, the cost of position around the point is made negative: well-placed texts help minimizing the total cost, but removing a poorly placed text does not penalize too much global quality. This criterion is weighted by label importance to prevent the system from removing important labels.

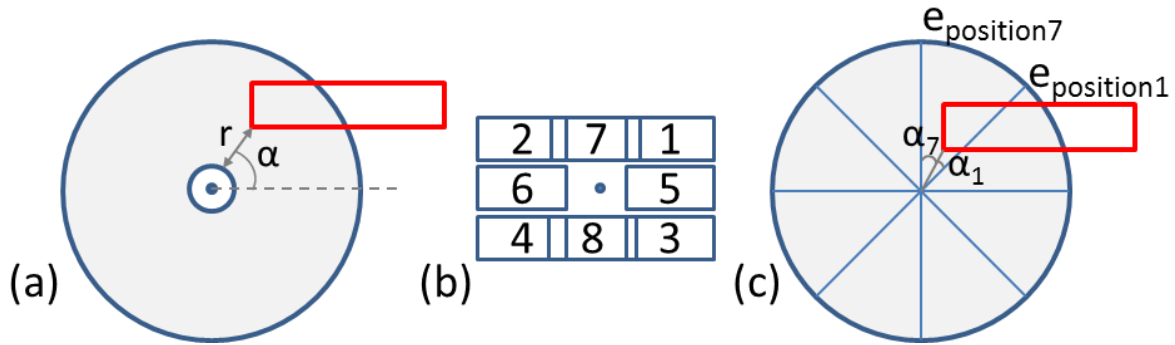


Fig. 1. Modelling label placement priorities in the optimization model: (a) the position is encoded with an angle value and a radius that guarantees that the label is not too close or too far from the related point feature; (b) the classical 8-position priority model is used, and (c) translated into a continuous model where priority is interpolated using the angles.

$$(Eq. 2) \quad \text{if } \alpha \in \left[\frac{\theta}{4}, \frac{\theta}{2}\right], e_{position} = e_{position7} + \frac{(e_{position1} - e_{position7})}{\theta/4} \cdot \left(\alpha_1 - \frac{\theta}{4}\right)$$

The unary energy for label position uses a continuous model for label position (Fig. 1). We chose this model first because of the used implementation of RJMCMC (Brédif & Tournaire 2012) that requires real intervals to pick into rather than 8 integer values, and because it gives more flexibility to move a label, in order to avoid label overlaps. Eq. 2 shows the principles of this energy computation for the case of a label position between positions 1 and 7, and the interpolation is carried out similarly for the seven other quadrants.

3. Preliminary Results

We carried out experiments on a cartographic dataset that was generalized for 1:25k maps, but we tried to display all the labels and the map at a smaller scale where the label bounding box covers more space and the need for a reduction of the number of labels is significant. The size of the text is intentionally large to force conflicts with many labels (Fig. 2).

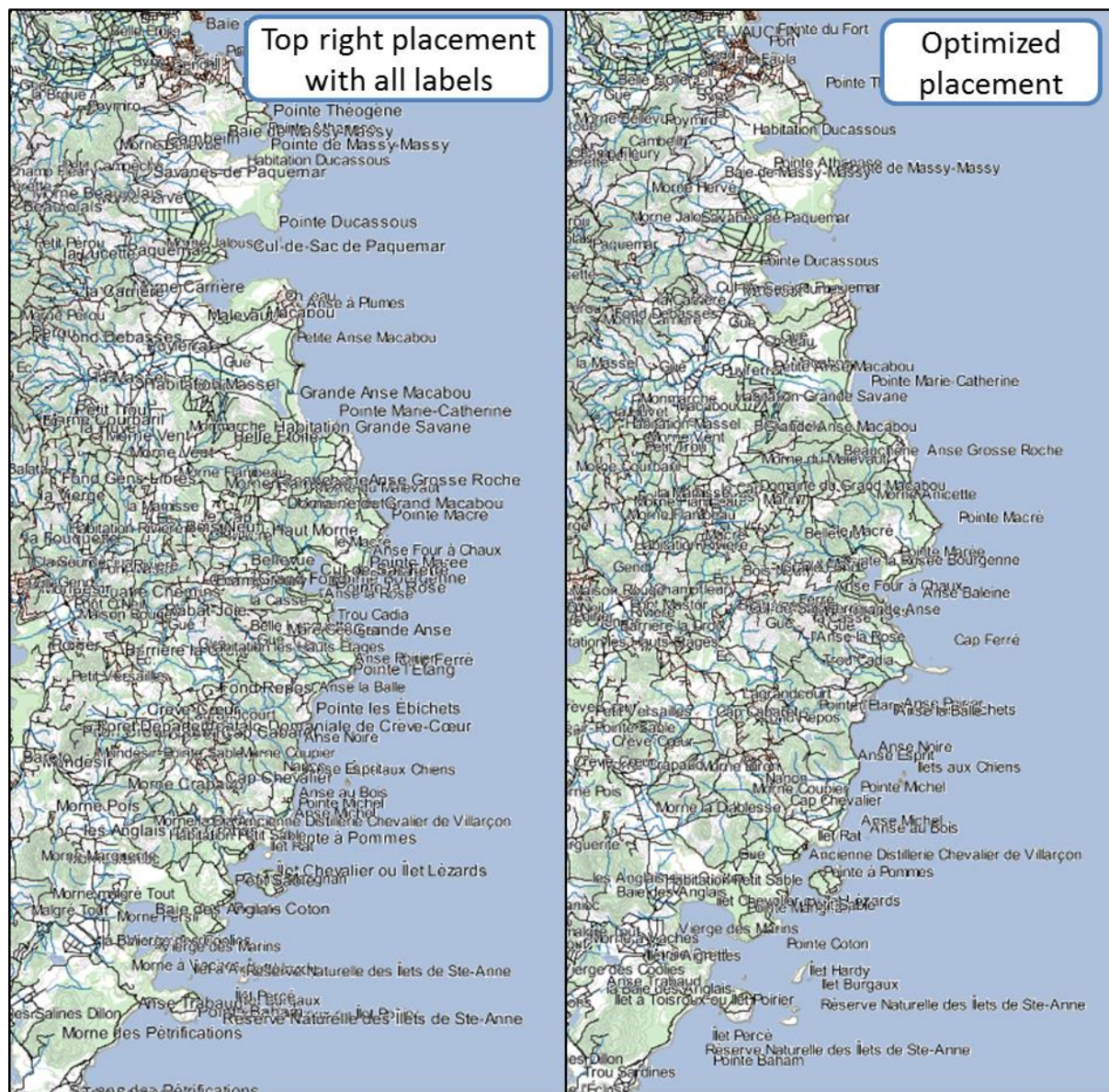


Fig. 2. Initial text placement with everything at the top right position (184 labels) and the optimized placement on a 1:25k map (114 labels left @IGN).

Fig. 3 shows a zoomed extract of the same dataset for a large number of iterations (60,000,000). The placement is clearly better with a significant number of labels removed but the result is not optimal. This can partly be explained by the caricatural size of the text, and its unique size: with only the main labels with this large size and smaller sizes to illustrate decreasing importance, there would be fewer conflicts to solve. However, this preliminary results shows that there is much to do to improve the proposed method.



Fig. 3. Zoomed extract of the test dataset for a large number of iterations (60,000,000) (@IGN).

3. Conclusions

The first results presented here are encouraging as the removal of some labels really improves map legibility. But there is still a lot to do to improve the method, by running more tests to tune the parameters and the criteria. For instance, it would be useful to compare the results of our RJMCMC optimization compared to a simulated annealing optimization with the same criteria and the same cost function. We also want to include text placement for linear (e.g. river names), and for areal (e.g. forest names) labels in the method that currently only considers point labels. We also plan to test the method on dense OpenStreetMap areas, because they may contain a huge number of named elements that can be displayed and the reduction of this number is mandatory. Finally, the optimization is quite slow for now, and we want to improve our heuristics (e.g. no need to search for overlaps with distant labels).

Acknowledgements

Our work is part of MapMuxing project, funded under grant ANR-14-CE24-0011-01 by French national agency for research.

References

- Barrault, M. (2001). A methodology for placement and evaluation of area map labels. *Computers, Environment and Urban Systems*, 25(1), 33–52.
- Brédif, M., Tournaire, O. (2012). libjmmcmc: An open-source generic c++ library for stochastic optimization. In *The XXII Congress of the International Society of Photogrammetry and Remote Sensing*.
- Green, P.J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika* 82(4), 711–732.
- Rylov, M.A., Reimer, A.W. (2014). A comprehensive multi-criteria model for high cartographic quality Point-Feature label placement. *Cartographica* 49(1), 52–68.